Thomas Wagner

Strickzeug

IEEE 802.11s: Mesh-WLANs auf Linux-Routern

Braucht ein Campingplatz oder Open-Air-Festival ein Funknetzwerk, gelangt ein einfacher WLAN-Access-Point schnell an seine Grenzen. Abhilfe versprechen Linux-Router mit OpenWRT, die ohne zusätzliche Kabel große WLAN-Wolken per IEEE 802.1s aufspannen. Das WLAN-Mesh-Verfahren vermeidet unnötige Übertragungen und spart Bandbreite, die Einrichtung unter OpenWRT benötigt aber noch etwas Handarbeit.

Will man große Flächen wie Campingplätze, Festival- oder Betriebsgelände mit einem Funknetz versorgen, kann man WLAN-Basisstationen darüber verteilen und sie per Kabel verbinden. Lohnt das nicht oder stehen unüberbrückbare Hindernisse im Weg, lässt sich das WLAN-Signal auch ohne Kabel bis in den letzten Winkel verteilen. Üblicherweise erledigen diese Aufgabe Repeater.

Flexibler und effizienter arbeiten vermaschte Funknetze (Mesh-WLAN) nach dem IEEE-Standard 802.11s: Während Repeater Datenpakete per Broadcast an alle WLAN-Teilnehmer weitergeben und damit eine Datenflut erzeugen, leitet ein per IEEE 802.11s vermaschter WLAN-Router die Pakete dank eines Routing-Verfahrens nur in Richtung der Empfänger weiter. Das vermeidet unnötige Übertragungen und spart Bandbreite. Außerdem verknüpfen sich 11s-Mesh-Router automatisch zu einem Funknetz – sie müssen nur eine gemeinsame Mesh-ID und den Funkkanal kennen. Repeater vergrößern hingegen nur das WLAN eines einzigen Access-Points, den sie über seine Hardware-Adresse (MAC) identifizieren.

Da die Kommunikation im Mesh-Netz über mehrere Knoten läuft, darf man bei der Geschwindigkeit keine Wunder erwarten: In einer Wolke aus drei Knoten sinkt die Übertragungsgeschwindigkeit mindestens um die Hälfte. Die Latenz hingegen steigt pro Hop um einige Millisekunden. Dabei kann es wie bei WLAN üblich zu Ausreißern nach oben kommen, die zu spürbaren Verzögerungen etwa bei interaktiven Diensten wie SSH oder Online-Spielen führen.

Für WLAN-Clients wie Notebooks oder Smartphones sieht ein 11s-Mesh-Netz aus wie ein gewöhnliches Funknetz: Besitzen alle 11s-Access-Points dieselbe Funkkennung (ESSID), können sich die Clients im Funkbereich der Router frei bewegen. Sinkt der Empfangspegel unter einen Schwellwert, bucht sich der Client um, wenn ein anderer Router besseren Empfang bietet.

Neben FreeBSD gehört Linux und damit auch das Router-Linux OpenWRT zu den ersten Betriebssystemen, die überhaupt Mesh-WLANs nach IEEE 802.11s aufbauen können. Obwohl Linux noch einige im Standard definierte 11s-Funktionen fehlen (siehe Kasten auf S. 173), läuft die Implementierung zwischen gleichen Kernelversionen bereits stabil.

11s-Mesh im Handbetrieb

Einschränkungen gibt es bei der WLAN-Hardware. Das WLAN-Kernel-Modul mac80211.ko emuliert auf Linux (und OpenWRT) bei Soft-MAC-WLAN-Karten den MAC-Layer, der die Mesh-Funktionen enthält. Stammt die WLAN-Karte von Atheros oder Ralink, sollte 11s funktionieren. Zeigt der Befehl `iw phy` unterhalb der Zeile „Supported interface modes“ die Angabe „mesh point“ an, lässt sich die WLAN-Karte auch als 11s-Meshknoten betreiben. Die meisten Linuxe bringen das Programm iw bereits mit, bei anderen lädt es der jeweilige Paketmanager nach.

Ein unverschlüsseltes 11s-Mesh-Netz lässt sich auf einem aktuellen Linux oder der zum Download bereitstehenden OpenWRT-Version Backfire vergleichsweise flott einrichten. Auf einem Desktop-Linux wie Ubuntu muss man zuvor einen laufenden Network Manager deaktivieren. Im folgenden Beispiel baut der Root-Nutzer auf einem Ubuntu-Linux (12.10) ein 11s-Mesh-Netz mit der ID „testmesh“ auf:

```
service network-manager stop
iw phy phy0 interface add mesh0 type \
mp mesh_id testmesh
iw dev mesh0 set channel 1
ifconfig mesh0 inet 192.168.0.1 netmask 255.255.255.0 up
```

Der im dritten Kommando angegebene WLAN-Kanal lässt sich frei wählen, die anschließend gesetzte IP-Adresse muss auf allen Mesh-Knoten zum gleichen Subnetz gehören. Laufen im Gerät mehrere WLAN-Karten, lassen sie sich über den Befehl `iw phy` auflisten. Nun ersetzt man im zweiten Kommando den Wert `phy0` durch die Geräte-Kennung. Einen zweiten Mesh-Rechner vernetzt man äquivalent, nur die IP-Adresse muss sich dabei auf jeden Fall von der des ersten unterscheiden. Anschließend lässt sich die Verbindung mit `iw dev mesh0 station dump` überprüfen.

Wenn der Befehl eine Ausgabe wie diese liefert,

```
Station ab:cd:ef:00:00:64 (on mesh0)
    inactive time: 670 ms
    mesh plink: ESTAB
    [...]
```

sollte man auch den zweiten Rechner per Ping erreichen können.

Verschlüsselt

Ohne zusätzliche Verschlüsselung lassen sich Verbindungen in diesem Mesh-Netz mit geringem Aufwand belauschen. Das verhindert derzeit nur die für 11s-Netze optionale Verschlüsselungstechnik Auth-SAE, die noch nicht zum Umfang der Linux-Distributionen gehört. Daher muss man die Software selbst übersetzen. Den Quelltext lädt das ohnehin installierte `wget` von Github herunter. Alle hier aufgeführten Downloads und URLs finden Sie über den Link am Ende des Artikels. Zudem sind einige Compiler, Übersetzungstools und Bibliotheken erforderlich. Die lassen sich so laden und einrichten:

```
sudo apt-get install build-essential cmake \
libnl-3-dev libnl-genl-3-dev libconfig8-dev \
libssl-dev libcrypto++-dev
```

Anschließend holt man die Auth-SAE-Quellen, entpackt sie, wechselt in das Archivverzeichnis authsae-master/ und startet die Übersetzung.

```
wget https://github.com/cozybit/authsae/
archive/master.zip -0 authsae.zip
unzip authsae.zip
cd authsae-master
make
```

Nach dem Kompilieren liegt im Verzeichnis ~/authsae-master/build/linux das Programm `meshd-nl80211`, das eine verschlüsselte Mesh-Verbindung aufbaut. Beispiele für die Konfiguration des Programms haben die Entwickler in der Datei ~/authsaemasterconfig/authsae.sample.cfg angegeben.

Auth-SAE funktioniert auf einigen Atheros-WLAN-Chipsätzen nur dann, wenn der Treiber mit der Option `nohwcrypt` geladen wurde. Das ist in den Voreinstellungen nicht der Fall, weshalb man den Treiber einmal erneut startet.

```
rmmod ath9k
insmod ath9k nohwcrypt=1
```

Die phy-Kennung der WLAN-Karte ändert sich dabei, so dass man sie, wie weiter vorn beschrieben, beim Kommando iw anpassen muss.

```
iw phy phy0 interface add mesh0 type mp
ifconfig mesh0 inet 192.168.0.1 netmask 255.255.255.0 up
meshd-nl80211 -c config/authsae.sample.cfg -i mesh0 -d
```

Maschen-Router

OpenWRT unterstützt seit der Version 10.03.1 (Backfire) Mesh-WLANs gemäß IEEE 802.11s, die sich in den fertigen OpenWRT-Firmwareversionen allerdings nur über die Kommandozeile und ohne Auth-SAE-Verschlüsselung einrichten lassen. Man kann sich jedoch mit den Erweiterungen der Hochschule RheinMain behelfen.

Damit lässt sich ein 11s-Mesh über OpenWRTs Web-Oberfläche LuCI konfigurieren und die Erweiterungen bringen auch die Mesh-Verschlüsselung Auth-SAE mit. Die Entwickler haben als Basis den Cisco-Router WRT160L verwendet, der WLANs über einen Atheros-Chipsatz mit dem Linux-Treiber ath9k aufspannt. Getestet wurden die angepasste OpenWRT-Fassung auch auf TP-Links Kleinst-Router MR3020, der ebenfalls mit einem Atheros-Chipsatz funkt. Geräte mit anderer WLAN-Hardware können jedoch mindestens unverschlüsselte Mesh-Netze aufbauen: Eine Liste OpenWRT-tauglicher Router findet sich im Wiki.

Damit der Aufbau eines verschlüsselten Mesh-Netzes funktioniert, muss man das gesamte OpenWRT-Paket selbst übersetzen. Da es dabei hier und da einige Klippen gibt, zeigen wir Ihnen hier, wie Sie den OpenWRT-Selbstbau vorbereiten und die Firmware übersetzen.

Das Hochschulprojekt nutzt die ältere SVN-Trunk-Version r32582. Man übersetzt sie zu einem OpenWRT-Firmware-Image am besten auf einem Linux – etwa in einer virtuellen Maschine wie Virtualbox. Für das Übersetzen braucht das Linux zusätzliche Programme, Compiler und Bibliotheken, die sich bei Debian und Ubuntu über den folgenden Befehl nachladen lassen:

```
sudo apt-get install build-essential \
subversion libncurses-dev zlib1g-dev gawk git
```

Danach legen Sie ein Projekt-Verzeichnis an und laden sich die OpenWRT-Quellen auf den Rechner:

```
mkdir openwrt_r32582
cd openwrt_r32582
svn co svn://svn.openwrt.org/openwrt/trunk/@32582
cd trunk
```

Jetzt fehlen noch die Erweiterungen des Hochschulprojektes …

```
mkdir dl
wget http://tinyurl.com/hsrm-mesh-addons \
--no-check-certificate \
-0 dl/hsrm.tbz
tar xf dl/hsrm.tbz
```

… sowie einige Einstellungen, die das Kommando `make defconfig` einleitet. Anschließend benötigen Sie noch die Quelltexte der OpenWRT-Web-Oberfläche LuCI und machen sie der Build-Umgebung bekannt:

```
./scripts/feeds update packages luci
./scripts/feeds install -a -p luci
./scripts/feeds install -d m libconfig
make download
```

Eine ausführliche Anleitung für die Einrichtung von OpenWRTs Buildsystem findet sich im Wiki des Open-Source-Projekts.

Der Befehl `make menuconfig` legt fest, auf welcher Hardware die Firmware laufen soll und welche Pakete in der Firmware stecken. Für den WRT160NL muss dazu die Konfigurationsvariable Target-System auf „Atheros AR7xxx/AR9xxx“, Subtarget auf „Generic“ und Target Profile auf den Wert Linksys WRT160NL eingestellt sein. Soll die Firmware auf dem TP-Link-Router MR-3020 laufen, bleiben Target-System und Subtarget wie

```
    Target System (Atheros AR7xxx/AR9xxx)  --->
    Subtarget (Generic)  --->
    Target Profile (TP-LINK TL-MR3020)  --->
    Target Images  --->
    Global build settings  --->
[ ] Advanced configuration options (for developers)  --->
[ ] Build the OpenWrt Image Builder
[ ] Build the OpenWrt SDK
```

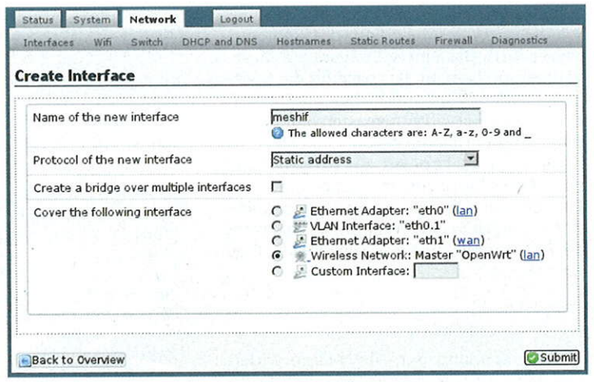
Über die drei oberen Punkte im Menü wählt man aus, auf welcher Router-Hardware das OpenWRT laufen soll.

Mesh-Status ermitteln

Auf der Root-Shell fragen die Programme iw und arp den Status des Mesh-Points ab. Unter OpenWRT heißt das Mesh-Interface meist wlan0 und bei MAPs wlan0-1.

Mesh abfragen

Befehl	Aufgabe, Funktion
iw phy	listet die WLAN-Fähigkeiten der Karte auf
iw dev wlan0 info	zeigt den Interface-Typ an
iw dev wlan0 station dump	zeigt alle Mesh-Points in Funkreichweite an
iw dev wlan0 mpath dump	zeigt alle vorhandenen Pfade mit Ziel-MAC-Adresse und Next-Hop-Adresse an
arp	Abbildung der IP-Adressen auf MAC-Adressen

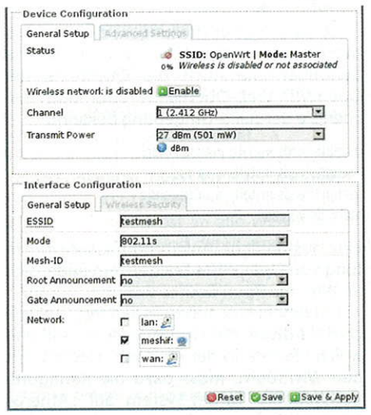


Das neue Mesh-Interface meshif gehört zum Wireless Network und setzt statische IP-Adressen voraus.

beim WRT160NL. Nur bei Target Profile stellt man „TP-LINK TL-MR3020" ein.

Außerdem müssen unter dem Menü „LuCI" der Punkt „luci-mesh" und im Unterverzeichnis „Collections" der Punkt „luCI" mit einem Stern markiert werden. So weist man das Buildsystem an, diese Pakete fest ins Image einzubauen. Steht der Cursor auf dem gewünschten Menüpunkt, drücken Sie dafür zweimal die Leertaste. Weiterhin benötigt die Firmware noch die Bibliothek libiw aus dem Menü „Libraries". Für die Verschlüsselung selektieren Sie die Pakete auth-sae und ath9-nohwcrypt im Menü „Network". Danach beenden Sie das Konfigurationsmenü, speichern die Vorgaben und stoßen mit dem Befehl `make world` den Firmware-Bau an.

Anschließend liegen im Unterverzeichnis bin/ar71xx mehrere Firmware-Dateien. Beim MR-3020 nehmen Sie die Datei openwrt-ar-71xx-generic-tl-mr3020-v1-squashfs-factory.bin und laden diese über das fürs Update zuständige Browser-Interface der Firmware auf das Gerät. Beim WRT160L benötigen Sie dafür die Datei openwrt-ar71xx-generic-wrt160l-squashfs-factory.bin. Läuft auf dem Gerät bereits OpenWRT, müssen Sie beim Aktualisieren auf jeden Fall den Haken bei „Keep Settings" entfernen. Ein Fehler im Firmware-Image würde ansonsten verhindern, dass der Router bootet. Installationshinweise zu den Router-Modellen finden sich im OpenWRT-Wiki.



Das WLAN-Interface schaltet man nun in den 802.11s-Modus und legt Funkkanal sowie Mesh-ID in der angepassten Web-GUI fest.

Für die Auth-SAE-Verschlüsselung muss bei Routern mit Atheros-WLAN-Karten das WLAN-Modul mit der Option `nohwcrypt=1` geladen werden: Für das Ath9k-Modul erledigt das das in der Firmware eingebaute Paket „ath9k-nohwcrypt" automatisch. Bei 11g-WLAN-Karten mit Atheros-Chip (Ath5k-Modul) ergänzen Sie die zusätzliche Option `ath5k nohwcrypt=1` auf dem Router in der Datei /etc/modules.d/*ath5k und starten anschließend das Gerät neu.

Mesh-Point

Die weitere Einrichtung des Mesh-WLAN läuft nun über die Web-Oberfläche des Routers, die Sie per Browser über die URL http://192.168.1.1 erreichen. Dort legen Sie unter dem Reiter „Network" mit „Add new interface…" eine neue WLAN-Schnittstelle an, geben ihr einen Namen (etwa meshif) und ordnen sie dem „Wireless Network" zu – der Button „Submit" speichert alles.

Anschließend setzen Sie auf der neuen Mesh-Schnittstelle eine feste IP-Adresse und schalten sie im Reiter „Wifi" vom Master-Mode in den Mesh-Mode (802.11s) um. Damit andere Mesh-Points das Gerät finden, müssen Mesh-ID und Funkkanal auf allen beteiligten Geräten übereinstimmen. Anschließend tragen Sie auch im Feld ESSID nochmals die Mesh-ID ein.

Nach dem Speichern der Einstellungen startet der Button „Enable" im Reiter Wifi den Mesh-Point. Beachten Sie, dass die Web-Oberfläche den Status des 11s-Netzes nicht immer korrekt ermittelt: Der Mesh-Point funktioniert selbst dann, wenn dort die Meldung „Wireless is disabled …" erscheint. Benachbarte, mit dem eigenen Gerät vermaschte Mesh-Points führt die Web-Oberfläche im Reiter Status unterhalb von „Associated Stations" auf.

Damit ein Mesh-Point zum Mesh-Access-Point (MAP) wird, bedarf es kleinerer Änderungen. Richten Sie zuerst das Interface meshif über den Reiter „Network", „Interfaces", „meshif", „Physical Settings", „Bridge interfaces" als Bridge ein.

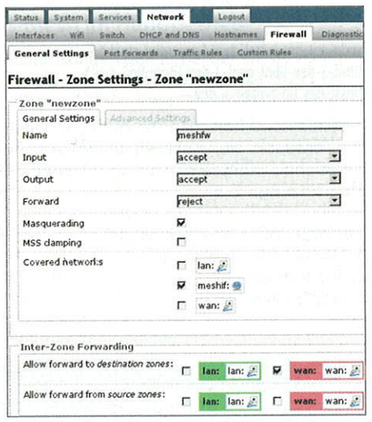
Anschließend fügen Sie über die Reiter „Network" und „Wifi" sowie den Add-Button ein klassisches WLAN hinzu, tragen eine ESSID ein und legen die Verschlüsselung dafür fest. Haben alle am Mesh beteiligten MAPs dieselbe ESSID und Passphrase, agiert das ganze Mesh-Netz wie ein einziger Access Point: Reißt der Kontakt zu einem MAP ab, verbindet sich ein WLAN-Client automatisch mit dem nächsten erreichbaren.

Nach der Eingabe der Einstellungen drücken Sie den Disable-Button, der sich daraufhin zum Enable-Button verwandelt: Ein weiterer Klick startet den neuen MAP.

Internet übers Mesh-Point-Portal

Der als Gateway arbeitende WLAN-Router wird mit zusätzlichen Einstellungen zum Mesh-Point-Portal: Laufen im Mesh bereits MAPs, muss auch das MPP als MAP konfiguriert sein. Ansonsten reichen zusätzliche Firewall-Einstellungen: Die Mesh-WLAN-Schnittstelle erhält eine eigene Firewall-Zone, was der Button „Add" im Abschnitt „Zones" auf den Reiter „Network", „Firewall" erledigt. Anschließend benennt man die neue Zone etwa mit „meshfw", aktiviert die Umsetzung privater in öffentliche IPv4-Adressen (Masquerading) und verbindet die Zone mit dem Mesh-Netz (siehe Bild unten).

Damit die WLAN-Clients über das Mesh-Point-Portal auch Netzwerkeinstellungen per DHCP erhalten, schalten Sie über die Reiter „Network", „Interfaces", „meshif" und den „Setup DHCP"-Button einen DHCP-Server hinzu. Achten Sie darauf, dass nur ein DHCP-Server im gesamten Mesh-Netz läuft. Sollen die Clients auch ins Internet gelangen, tragen Sie in der DHCP-Einrichtung auch die IP-Adresse des Mesh-Point-Portals (MPP) als Gateway ein.



Eine zusätzliche Firewall-Zone verbindet das Mesh-Netz auch mit dem Internet-Zugang des Routers.

Verschlüsseltes Mesh-Netz

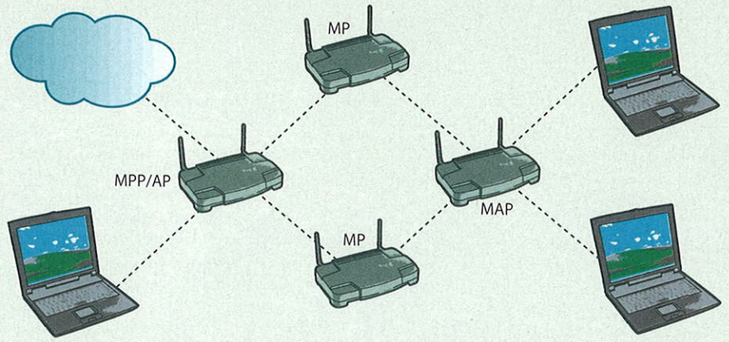
Auch für die Verschlüsselung des Meshs braucht es nur noch wenige Klicks: Ähnlich wie bei klassischen Funknetzen wählen Sie über den Reiter „Wireless Security" in der Web-Oberfläche den Wert „Auth-SAE" aus. Im Feld „Key" hinterlegen Sie nun die für die Verschlüsselung nötige Passphrase, die möglichst lang und kryptisch sein sollte – den Rest erledigen OpenWRT-Skripte, die nach dem Speichern alle Vorgaben setzen.

Zurzeit muss man noch etwas basteln, um ein 11s-Netz aufzuspannen. 11s-Mesh-Netze laufen mit Linux zwar bereits stabil, doch noch beherrschen OpenWRT und seine Linux-Verwandtschaft nicht alle Funktionen des Protokolls. (rek)

www.ct.de/1304170

Mesh-Netzwerke

Mesh-Funknetze kennen in der Regel keinen Master-Knoten, sie verwalten sich selbst: Jedes Gerät baut zu anderen, in Funkreichweite stehenden Knoten eine direkte Verbindung auf. Dabei entscheidet ein Routingprotokoll, welchen Weg die Daten zum Zielknoten nehmen. Nicht benachbarte, also außer Reichweite befindliche Mesh-Knoten kommunizieren über Zwischenstationen, die deren Daten weiterreichen. Mesh-Knoten können hinzukommen oder wegfallen und ihre Positionen ändern. Das Routing-Protokoll berücksichtigt diese Veränderungen und passt die Datenpfade automatisch an.



In einem 11s-Mesh vernetzen sich alle Router untereinander. Aus Sicht eines Notebooks erscheinen die MAPs wie normale Access Points. Die Notebooks bewegen sich frei, solange sie in Funkreichweite eines MAPs sind, haben sie auch eine Netzwerkverbindung.

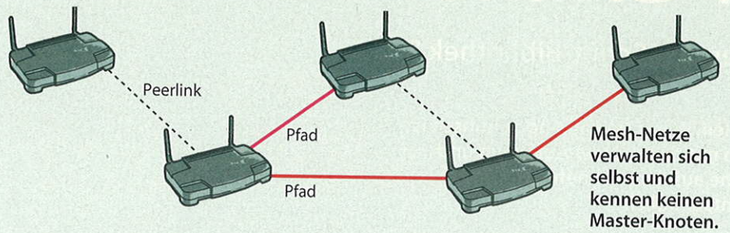
802.11s in der Wildnis

Inzwischen gehört die Mesh-WLAN-Erweiterung 802.11s zur WLAN-Spezifikation 802.11-2012 des Institute of Electrical and Electronics Engineers (IEEE). Gegenüber anderen Mesh-Verfahren zeichnet es sich dadurch aus, dass das Routing auf MAC-Ebene (Layer 2) stattfindet und nicht auf IP-Ebene (Layer 3) – wie bei B.A.T.M.A.N oder OLSR. Das verspricht mehr Geschwindigkeit.

WLAN-Netze gemäß IEEE 802.11s verwenden statt einer Funknetz-Kennung (Extended Service Set IDentifier) eine Mesh-ID: Alle Netzwerkknoten, die auf derselben Frequenz funken und dieselbe Mesh-ID verwenden, bauen zueinander Peerlinks auf. Die Kommunikation über diesen Peerlink lässt sich verschlüsseln, wobei man prinzipiell Verschlüsselungsmethoden wie WPA oder WPA2 einsetzen kann. Die Linux-Implementierung beherrscht derzeit nur das auf Mesh-Netze spezialisierte Auth-SAE (siehe c't-Link).

Routing

Normalerweise routen 11s-Mesh-Netze mit dem Hybrid Wireless Mesh Protocol (HWMP), andere Protokolle lassen sich aber nachrüsten. HWMP kennt einen proaktiven und einen reaktiven Modus. Im reaktiven Modus erfragt das Protokoll einen Pfad erst dann, wenn Daten zu einem bestimmten Ziel geschickt werden. Im proaktiven Modus arbeitet ein Mesh-Point als Root-Knoten und gibt sich mit einem Root-Announcement im Mesh bekannt. Die anderen Netzwerkteilnehmer erstellen daraufhin einen Pfad zum Root-Knoten, sodass eine sternförmige Topologie entsteht, über die sich alle Mesh-Knoten erreichen können. Dies widerspricht dem oben beschriebenen Mesh-Konzept (kein Master-Knoten) nur auf den ersten Blick: Fällt der Root-Knoten aus, fehlt auch das Root-Announcement und das Netzwerk wechselt automatisch in den reaktiven Modus. Unter Linux funktioniert derzeit allerdings nur der reaktive Modus.



Mesh-Netze verwalten sich selbst und kennen keinen Master-Knoten.

Komponenten

In einem 11s-Mesh-Netz kann jeder Knoten verschiedene Rollen übernehmen: Ein Mesh-Point (MP) empfängt, sendet und reicht Daten gemäß dem Routingprotokoll weiter. Sollen auch andere, nicht 11s-taugliche WLAN-Geräte wie Smartphones oder Notebooks über das Mesh-Netz Daten übertragen, braucht es MPs, die zusätzlich als Access-Point arbeiten (Mesh-Access-Points, MAP).

Wenn MPs oder MAPs auch als Gateway zu anderen Netzen agieren, nennt man sie Mesh-Point-Portal (MPP). Im IEEE-Standard 802.11s heißt der Mesh-Point inzwischen Mesh-Station, der Mesh-Access-Point und das Mesh-Point-Portal heißen dort Mesh-Portal respektive Mesh-Gate. Die Linux-Entwickler sind bisher bei der alten Terminologie geblieben, also MP, MAP und MPP.

Geschlossene Mesh-Ebene

Die MAPs und MPs arbeiten auf unterschiedlichen Ebenen: Clients können von einem MAP aus keinen MP erreichen und umgekehrt. In einem Netz mit MPs und MAPs leiten die MPs ausschließlich Daten weiter. Klassische WLAN-Clients erhalten über einen MAP nur dann Zugang zu anderen Netzen, wenn das Gateway sowohl als MAP als auch MPP eingerichtet wurde.

c't